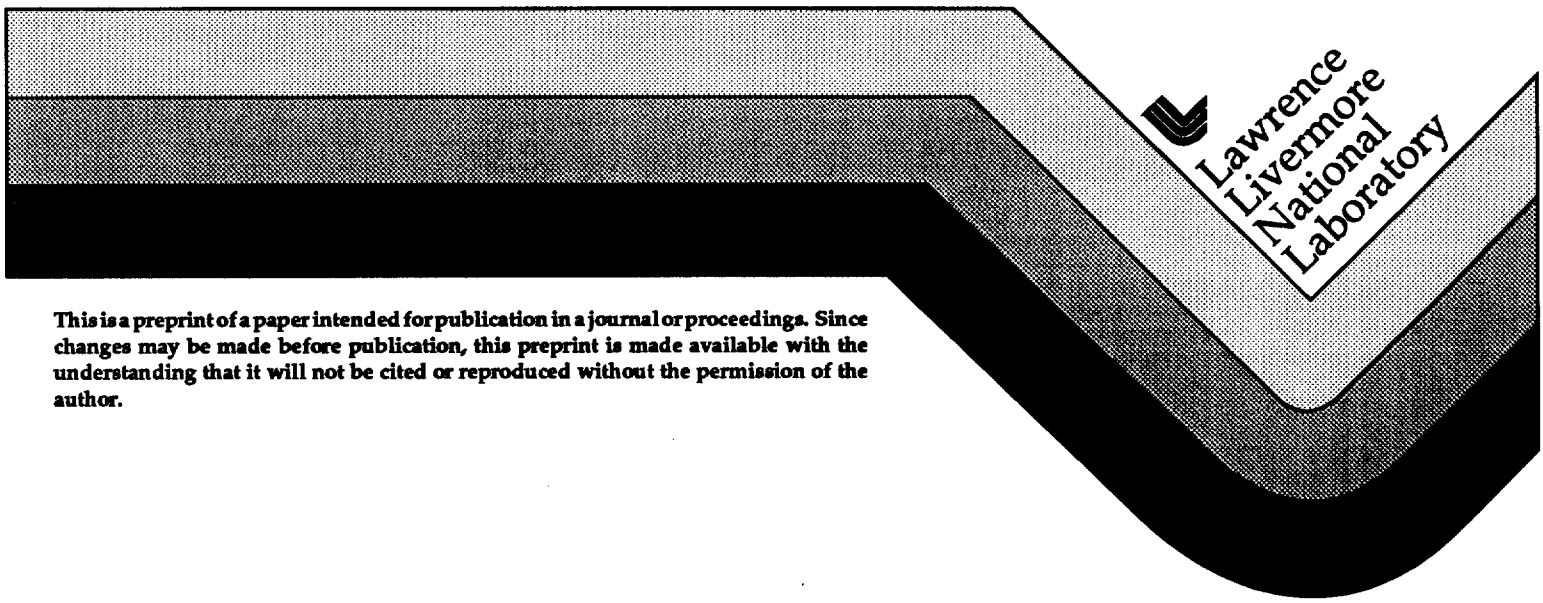
UCRL-JC-124449
PREPRINT

# Efficient Transformations from Geodetic to UTM Coordinate Systems

R. M. Toms

This paper was prepared for submittal to the
Fifteenth Workshop on Standards for the Interoperability
of Distributed Simulations
Orlando, FL
September 16-20, 1996

August 7, 1996




**This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.**

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

# Efficient Transformations From Geodetic to UTM Coordinate Systems

**Ralph M. Toms**
**Advanced Simulations Projects Manager**
**Conflict Simulation Laboratory**
**Lawrence Livermore National Laboratory**

**P. O. Box 808, L-184**
**Livermore, California 94551**
**e-mail address: ralph@tomtom.llnl.gov**
**Telephone: 510-423-9828**
**FAX: 510-423-2492**

KEYWORDS

Coordinate Conversion/System,
Correlation,
Datum Transformation,
Modeling and Simulation Interoperability

## ABSTRACT

The problem of efficiently performing transformations from geocentric to geodetic coordinates has been addressed at previous DIS Workshops. This paper extends the work presented at the Fourteenth DIS Workshop. As a consequence of the new algorithm for geocentric to geodetic coordinate conversion, a subsequent conversion to Universal Transverse Mercator coordinates is made considerably more efficient. No additional trigonometric or square root evaluations are required and accuracy is not degraded.

## INTRODUCTION

The Distributed Interactive Simulation (DIS)[1,2] environment requires frequent transformations between the DIS standard coordinate system and other coordinate systems used at DIS nodes. When entities are maneuvering, this requirement induces substantial computational loads on both the sending and receiving nodes. One of the more stressing situations occurs when a point in the standard geocentric system is transformed to geodetic coordinates and subsequently to Universal Transverse Mercator (UTM) coordinates.

New algorithms have been developed that substantially increase the efficiency of the transformation from geocentric to geodetic coordinates[3,4]. This note describes an extension of these algorithms that increases the efficiency of the transformation from geodetic to UTM coordinates. Results of this note should be applicable to applications other than DIS where efficient transformations of the type addressed are required.

### Background

The DIS program maintains a set of recommended standard algorithms for coordinate transformations at the Visual Systems Laboratory of the Institute for Simulation and Training (IST), University of Central Florida. Proposed new algorithms are sent to IST for testing in terms of efficiency and accuracy. The best of the current implementations for obtaining UTM coordinates is based on the transformation equations contained in a USGS Bulletin "Map Projections Used by the U. S. Geological Survey"[5]. These equations are written in a form that is readable, preserves accuracy and applies to all situations. However, the direct implementation of the equations, does not lead to an efficient implementation.

In the DIS environment, the conversion to UTM coordinates almost always follows a conversion from geocentric coordinates to geodetic coordinates. As a consequence, the trigonometric functions of latitude are available from the algorithm of references 3 and 4 and need not be re-computed. Likewise, the radius of curvature is also available. In addition, the UTM transformation equations involve trigonometric functions of integer multiples of the latitude. By utilizing multiple angle identities, these evaluations can done without evaluating any trigonometric functions. These observations permit the transformation equations to be written in a form that allows maximal factorization of common factors and requires no square roots or trigonometric calculations. The resulting procedure is estimated to be on the order of fifty percent faster than the existing recommended DIS procedure.

## The Transverse Mercator Coordinate System

The Transverse Mercator system is a conformal (angle preserving) projection of points relative to a geodetic Earth model onto a plane tangent to the geodetic surface at the intersection of the central meridian and the equator. Under this transformation both the central meridian and the equator map into straight lines. No other meridian or parallel maps to a straight line under this projection. While conformal mappings preserve angles other distortions are introduced as shown in Figure 1.

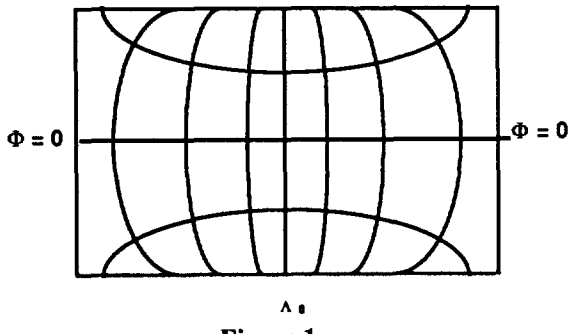



**Figure 1**

**Meridians and parallels in the Transverse Mercator coordinate system.**

As can be seen in the figure, the distortions are small near the origin and increase away from it. To minimize this distortion, the concept of a Universal Transverse Mercator (UTM) projection is introduced.

## The Universal Transverse Mercator (UTM) Coordinate System

UTM coordinates are defined in the following manner. The Transverse Mercator projection is modified by introducing a set of longitudes spaced six degrees apart on the equator. Each of these points is used as a local origin for a Transverse Mercator projection. In essence, the central meridian is replaced by a set of meridians equally spaced around the equator. This spacing corresponds to a set of 60 UTM zones. In any one zone the Transverse Mercator projection becomes localized and distortions are minimized.

Transverse Mercator projections are defined with respect to a geodetic earth model, the associated rotating geocentric coordinate system and the corresponding geodetic coordinate system. A number of reference geodes have been used in astrogeodetic work[6]. These all have the form:

$$(1) \qquad (X/a)^2 + (Y/a)^2 + (Z/c)^2 = 1.$$

Figure 2 below depicts the geometry of the geocentric (Cartesian) system and the geodetic system in three dimensions. The geocentric coordinates of a point P are (X,Y,Z) and the corresponding geodetic coordinates of P are ($\phi$,$\lambda$,h) where $\phi$ is latitude, $\lambda$ is longitude and h is the height above the reference ellipsoid. The line connecting the Z axis to P is orthogonal to the tangent plane at the point $P_e$.

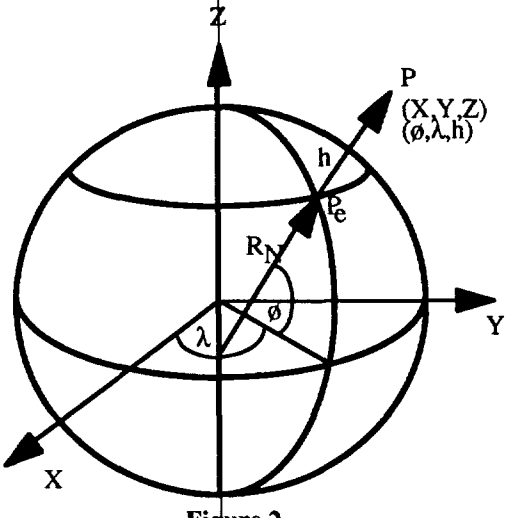



**Figure 2**

**Cartesian and Geodetic Systems**

The transformation from geodetic to geocentric coordinates is straightforward[6] and is given by:

(2) $X = (R_N+h) \cos\phi \cos\lambda,$

(3) $Y = (R_N+h) \cos\phi \sin\lambda,$

(4) $Z = (R_N c^2 / a^2 + h) \sin\phi.$

where $R_N$ is the radius of curvature of the prime vertical and is given by

(5) $R_N = a / [1 - [\sin^2\phi] (a^2 - c^2) / a^2 ]^{1/2}.$

The inverse transformation is expressible in closed form,[7] but this generally results in an inefficient implementation. An accurate and efficient iterative algorithm is contained in references 3 and 4.

The longitude $\lambda$ is given by

(6) $\lambda = \tan^{-1} (Y / X),$

where $-\pi \le \lambda \le \pi.$

Since the two minor axes of the geode are equal, the projection of the geode on any meridional plane defines an ellipse. The eccentricity of the ellipse is defined by

(7) $\varepsilon^2 = ( a^2 - c^2 ) / a^2.$

It is convenient to define an additional constant $\varepsilon'$ squared by

(8) $\varepsilon'^2 = ( a^2 - c^2 ) / c^2.$

The flattening ratio f of the ellipse is defined by

(9) $f = (a - c) / a.$

Reference 7 contains a good discussion of the process for transforming from geodetic to Transverse Mercator to UTM coordinates. In the appendix, the equations used in reference 5 and in the DIS standard code are summarized using notation consistent with references 3 and 4.

## EFFICIENT TRANSFORMATION FROM GEODETIC TO UTM COORDINATES

The transformations given in the Appendix involve the sine of multiple angles. Most of the implementations of these equations and similar equations contained in reference 7 call the sine function repeatedly to evaluate the sine of each multiple angle. This is relatively costly in terms of processing time. The use of the well known multiple angle identities avoids the unnecessary calls. However, multiple angle identities are not unique and some care is needed to pick a form that maximizes the factorability of the resulting equations. The variables used in this section are defined in the appendix. Accordingly the following are used:

(10) $\sin 2\phi = 2 \sin\phi \cos\phi,$

(11) $\cos 2\phi = 1 - 2 \sin^2\phi,$

(12) $\sin 4\phi = 2 \sin 2\phi \cos 2\phi,$

(13) $\sin 6\phi = 2 \sin 2\phi \cos^2 2\phi + \sin 2\phi (1 - 2 \sin^2 2\phi ).$

Using equations (12) and (13), equation (35) of the appendix can be written

(14) $M = a \{A_0 \phi - \sin 2\phi [A_2 - 2 \cos 2\phi (A_4 - A_6 (1 - \sin^2 2\phi ))]\}.$

As previously noted, when UTM coordinates are needed in DIS applications they almost always follow a transformation from geocentric to geodetic coordinates. If the algorithms of reference 4 or 5 are used, the following are already computed using just square roots: $\sin\phi$, $\cos\phi$, $\sin^2\phi$, $\cos^2\phi$, $\tan\phi$ and $R_N$. These values can be used in an integrated routine that transforms from geocentric to geodetic, and on option, to UTM coordinates. Coupled with (10) to (13) the computation of M reduces to simple algebraic operations.

As shown in the appendix, the Transverse Mercator coordinates of a point, $x_{TM}$ and $y_{TM}$, are computed prior to computing the UTM coordinates. The equations for $x_{TM}$ and $y_{TM}$ are polynomials in A. If these equations were coded as shown, optimizing compilers would automatically factor out common factors in A. Unfortunately, many of the existing implementations compute the powers of A and store them as constants prior to coding the polynomial. As a result, the compiler may not be able to recognize the common factors. To insure that the factorization occurs the equations should be written as follows,

(15) $x_{TM} = k_0 R_N A\{1 + A^2 [(1 - T + C)/6 + (5 - 18T + T^2 + 72C - 58 \varepsilon'^2)A^2/120]\}.$

(16) $y_{TM} = k_0 \{M + R_N \tan\phi A^2 [1/2 + A^2((5 - T + 9C + 4C^2 ) / 24 +$

$$(61 - 58T + T + 600C - 330\,\varepsilon'^{2})A^2 / 720)]].$$

When a particular geode is used, further economies can be obtained by reformatting (14), (15) and (16). This reformatting also will insure that constants are not repeatedly computed. Accordingly, let

(17) $B_0 = ak_0A_0,$

(18) $B_2 = 2ak_0A_2,$

(19) $B_4 = 4ak_0A_4,$

(20) $B_6 = 4ak_0A_6,$

(21) $B_7 = B_4 - B_6,$

(22) $B_8 = 4B_6,$

(23) $v = \sin\phi\cos\phi,$

(24) $u = \sin^2\phi,$

(25) $M_k = k_0 M.$

With these definitions

(26) $M_k = B_8\phi - v\{B_2 - (1-2u)[B_7 + v^2B_8]\},$

(27) $x_{TM} = R_N A\{k_0 + A^2[(1 - T + C)k_0/6 +$

$(5 - 58\,\varepsilon'^{2} - 18T + T^2 + 72C)A^2 k_0/120]\},$

(28) $y_{TM} = \{M + R_N \tan\phi\, A^2[k_0/2 +$

$A^2((5 - T + 9C + 4C^2)k_0/24 +$

$(61 - 330\,\varepsilon'^{2} - 58T + T^2 + 600C)A^2k_0/720)]\},$

where the terms $k_0/2$, $k_0/6$, $k_0/24$, $k_0/120$, $k_0/720$, $5 - 58\,\varepsilon'^{2}$ and $61 - 330\,\varepsilon'^{2}$ are pre-computed global constants.

## RESULTS

Formal timing studies that compare the new formulation to the DIS standard have not yet been completed. The DIS standard code based on reference 5 was obtained from the Visual Systems Laboratory of the Institute for Simulation and Training, University of Central Florida during February of 1996. Operations counts were obtained for each distinct call of the routine (global constants were ignored). Similarly, operations counts were obtained for the new algorithm (i.e. equations (23),(26),(27),(28),(32),(33),(34),(40), (42),(44) and (46).

The results are summarized in Table 1 below.

**Table 1. Operations count for standard and new algorithm.**

| | DIS/IST | New Algorithm |
|---|---|---|
| Multiply | 40 | 35 |
| Add | 23 | 22 |
| Divide | 6 | 0 |
| Square Root | 1 | 0 |
| Sine or Cosine | 5 | 0 |

In the presentation of reference 3 at the 13th DIS Workshop experimental data were presented indicating that square root function calls take approximately the same processing time as five floating point multiplies on machines equipped with floating point units. Similarly, the sine or cosine take the same time as five to twelve floating point multiplies, depending on the particular machine. If all the arithmetic operations in Table 1 are assumed to take the same time as a floating point multiply, estimates of relative execution time can be made.

The above analysis shows that the DIS/IST algorithm takes between 99 and 134 operations while the new algorithm takes 57 operations. The percent improvement is defined in terms of old time minus the new time divided by the old time. In such terms the estimated improvement ranges from 42 to 57 percent. These estimates may be conservative in that the DIS/IST procedure takes 46 long operations (multiply or divide) while the new algorithm takes only 35 and generally the long operations are more expensive in terms of computer time.

Howard Lu of SAIC in Boston has implemented an early version of the new algorithm and has found improvements in the same range as above. The author is indebted to Howard for communicating this information.

## CONCLUSIONS

The algorithm proposed above is most suitable for DIS applications and may have utility in other compute intensive environments. The procedure should be readily adaptable to the standard UTM transformations that are contained in the Military Handbook (reference 7). These differ slightly from the transformations appearing in reference 5, in that more terms are used in some of the series approximations.

## APPENDIX

The transformations from geodetic to Transverse Mercator coordinates are taken directly from reference 5. The mathematical form of the equations shown below corresponds exactly to that used in the reference. A few variable name changes are employed to make these equations correspond to the notation used in references 3 and 4.

Transformation from geodetic to Transverse Mercator

A point with geodetic coordinates $(\phi,\lambda,h)$, where the angles are in radians, has Transverse Mercator coordinates $(x_{TM},y_{TM})$ given by,

(29) $$x_{TM} = k_0 R_N \{A +(1 - T+ C)A^3/6 + (5 - 18T + T^2 + 72C - 58\varepsilon'^2)A^5/120\},$$

(30) $$y_{TM} = k_0 \{M + R_N \tan\phi (A^2/2 + (5 - T+9C +4C^2) A^4/24 + (61 -58T + T+600C -330\varepsilon'^2) A^6/720]\},$$

where $k_0$ is the point scale factor and

(31) $\lambda_0$ = central meridian in radians,

(32) $$A = (\lambda - \lambda_0)\cos\phi,$$

(33) $$T = \tan^2\phi,$$

(34) $$C = \varepsilon'^2\cos^2\phi,$$

(35) $$M = a(A_0\phi - A_2\sin 2\phi + A_4\sin 4\phi - A_6\sin 6\phi),$$

(36) $$A_0 = 1 - \varepsilon^2/4 - 3\varepsilon^4/64 - 5\varepsilon^6/256,$$

(37) $$A_2 = 3\varepsilon^2/8 + 3\varepsilon^4/32 + 45\varepsilon^6/1024,$$

(38) $$A_4 = 15\varepsilon^4/256 + 45\varepsilon^6/1024,$$

(39) $$A_6 = 35\varepsilon^6/3072.$$

The altitude h does not enter into the transformation to Transverse Mercator or to UTM coordinates.

Transformation from Transverse Mercator to UTM

The UTM projection differs from the Transverse Mercator projection as follows:

1. The longitude of each central meridian, in degrees, is given by $3 + 6n$, $n = 0, 1, ...., 59$.
2. The point scale factor $k_0$ along the central meridian is 0.9996.
3. The northing coordinate $y_{UTM}$ has an origin 0 at the equator for points in the northern hemisphere. The southing coordinate $y_{UTM}$ has an origin equal to 10,000,000. meters at the equator for points in the southern hemisphere. Southing decreases as points move in the direction of the south pole.
4. The easting coordinate $x_{UTM}$, has its origin equal to 500,000. meters at the central meridian.

If the longitude $\lambda$ is expressed in radians, the UTM zone is found from:

(40) $$z = \text{Greatest Integer} \le (31+180\lambda/(6\pi)),$$

for $0 \le \lambda < \pi$.

(41) $$z = \text{Greatest Integer} \le (180\lambda/(6\pi)-29),$$

for $\pi \le \lambda < 2\pi$.

For each zone, the central meridian $\lambda_0$, expressed in radians, is given by:

(42) $$\lambda_0 = (6z - 183)\pi/180 \quad \text{for } z \ge 31,$$

or

(43) $$\lambda_0 = (6z + 177)\pi/180 \quad \text{for } z \le 30.$$

Once the zone has been identified and the central meridian computed, UTM coordinates are

computed, in meters, from Transverse Mercator coordinates by

(44) $X_{UTM} = X_{TM} + 500{,}000.$

For points in the northern hemisphere,

(45) $Y_{UTM} = Y_{TM}.$

For points in the southern hemisphere,

(46) $Y_{UTM} = Y_{TM} + 500{,}000.$

**REFERENCES**


1. *The DIS Vision, A Map to the Future of Distributed Simulation*, Institute for Simulation and Training, University of Central Florida, Orlando Florida, May 1994.

2. *STRICOM/DMSO Standard for Distributed Interactive Simulation-Application Protocols, Version 2.1, Second Draft*, Institute for Simulation and Training, University of Central Florida, Orlando Florida, 1995.

3. Toms, R. M., *An Efficient Algorithm for Geocentric to Geodetic Coordinate Conversion*, Proceedings on Standards for the Interoperability of Distributed Simulations, Volume II, Institute for Simulation and Training, Orlando, FL., pp 635-642, September 1995.

4. Toms, R. M., *An Improved Algorithm for Geocentric to Geodetic Coordinate Conversion*, Proceedings on Standards for the Interoperability of Distributed Simulations, Volume I, Institute for Simulation and Training, Orlando, FL., pp 1135-1144, February 1996.

5. Snyder, J. P., *Map Projections Used by the U. S. Geological Survey*, Geological Survey Bulletin; 1532, U. S. Government Printing Office, Washington D. C., 1982.

6. Borkowski, K. M., *"Transformation of Geocentric to Geodetic Coordinates Without Approximations," Astrophys. Space Science*, 139, pp 1-4, 1987 and 146, pp 201, 1988.

7. Military Handbook, *Datums, Projections, Grids and Common Coordinate Systems, Transformation Of*, US Department of Defense, MIL-HDBK-600008, January 1996.



*This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.